3DCFS: Fast and Robust Joint 3D Semantic-Instance Segmentation via Coupled Feature Selection

Liang Du[†1], Jingang Tan[†2], Xiangyang Xue[3], Lili Chen[2],
Hongkai Wen[1,4], Jianfeng Feng[1], Jiamao Li[2] and Xiaolin Zhang[2]

***Abstract*— We propose a novel fast and robust 3D point clouds segmentation framework via coupled feature selection, named 3DCFS, that jointly performs semantic and instance segmentation. Inspired by the human scene perception process, we design a novel coupled feature selection module, named CFSM, that adaptively selects and fuses the reciprocal semantic and instance features from two tasks in a coupled manner. To further boost the performance of the instance segmentation task in our 3DCFS, we investigate a loss function that helps the model learn to balance the magnitudes of the output embedding dimensions during training, which makes calculating the Euclidean distance more reliable and enhances the generalizability of the model. Extensive experiments demonstrate that our 3DCFS outperforms state-of-the-art methods on benchmark datasets in terms of accuracy, speed and computational cost.**

I. Introduction

3D scene understanding based on LiDAR, RGB-D and stereo cameras has received increasing attention from both academia and industry because of its critical role in robotic scene perception, robotic manipulation and autonomous driving [1], [2]. Instance and semantic segmentation are the most widely used tasks in this research field. Building on the great success achieved in recent years [3]–[6] for each single task, joint learning methods for both tasks [7], [8] have opened up a more effective way to improve performance and promote further developments.

The two tasks have some common ground that can be associatively utilized to boost their performance. For example, points with different classes must be from different instances, and points from the same instance must be of the same class. The simplest but most naive methods to jointly perform instance and semantic segmentation are progressively using the predicted semantic labels to further cluster instances or utilizing the predicted instance results as prior knowledge for semantic segmentation. Nevertheless, using the unreliable upstream prediction as prior information may affect the downstream task; consequently, such approaches may be suboptimal. Another approach is to directly combine the high-level features of both tasks to perform information integration [7]. Although semantic and instance segmentation share the same goal of detecting specific informative regions, they have different individual learning orientations, and some part of the information they contain may be contradictory. Instance segmentation focuses on extracting point features from different objects to distinguish them, while the features extracted by semantic segmentation are used to classify points with different categories; as a result, the features of both tasks definitely contain different task-oriented parts. Therefore, feature selection is an essential step in the reciprocal process.

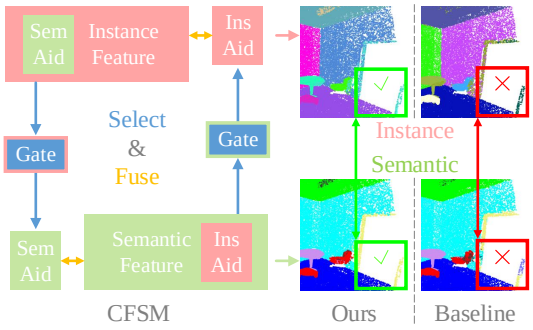

Fig. 1. An illustration of CFSM and a qualitative comparison of our method and the baseline. The baseline is a traditional multitask framework without CFSM, as introduced in Section III. Our framework via coupled feature selection is able to exploit and integrate the reciprocal information from both tasks based on gate mechanism to boost their performance, as shown in the marked regions.

Actually, such selection within a mutually aided process is consistent with human scene perception. Semantic and instance segmentation are the most important visual tasks in human scene perception. For humans, semantic perception mainly abstracts the advanced semantic features from the objects in the scene, while instance segmentation pays more attention to exploiting the primary features. These two processes can complement each other. Specifically, the mapping from advanced features to primary features can be beneficial for instance segmentation. For example, if we know the category of an object, we will obtain the blur shape information (primary features), which can help to correct errors in the instance segmentation result if it is difficult to see the

† The first two authors contributed equally to this work.

* This work was supported by the 111 Project (NO.B18015), the National Natural Science Foundation of China (No.91630314), the key project of Shanghai Science & Technology (No.16JC1420402), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01) and ZJLab.

[1]Liang Du, Hongkai Wen, Jianfeng Feng are with the Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, China, Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence (Fudan University), Ministry of Education, China. duliang@mail.ustc.edu.cn, jffeng@fudan.edu.cn

[2]Jingang Tan, Lili Chen, Jiamao Li and Xiaolin Zhang are with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China.

[3]Xiangyang Xue is with the School of Computer Science, Fudan University, China.

[4]Hongkai Wen is with the Department of Computer Science, University of Warwick, UK.

whole object because of environmental light. By contrast, if we know the primary features of objects of the same category that are unknown, we can establish links between those that belong to the same category, which can help us to quickly and accurately accomplish the semantic segmentation. Consequently, these two processes are not independent but coupled. However, humans are rarely disturbed by such different task-oriented information because the human brain has the capability to quickly and adaptively select useful information instead of the entire set of information. This characteristics of human scene perception inspired us to build a multitask-coupled framework to simulate the information-selection process of human scene perception via gate control units for robotics. The coupled and gate-based training pipeline is shown in Figure 1. For the encoder, we use the PointNet/PointNet++ utilized by [7], [9]. For the decoder, we investigate a novel coupled feature selection module (CFSM) that contains two coupled instance-to-semantic and semantic-to-instance streams to extract useful information while filtering useless information.

Our 3DCFS uses the Euclidean distance to calculate the similarity of different embeddings among all points for clustering. However, the Euclidean distance is sensitive to the magnitudes of different embedding dimensions, which makes the clustering result depend on only a small number of dimensions and reduces the generalizability of the model. We therefore propose a novel loss function, named $\mathcal{E}_{EMED}$, that helps the model learn to maintain equilibrium among the magnitudes of the instance embedding dimensions. In summary, the main contributions of our work are as follows:

- We propose a fast, yet effective end-to-end point clouds segmentation framework that simultaneously performs semantic and instance segmentation inspired by the human scene perception process.
- We introduce a novel coupled feature selection module (CFSM) to exploit the potential reciprocal information in semantic and instance segmentation tasks to seamlessly fuse the heterogeneous features, allowing these two tasks to benefit from each other.
- We design a novel loss for instance segmentation in 3DCFS, which helps the model learn to balance the magnitudes of the embedding dimensions to maintain the stability of the Euclidean distance calculation during training.
- We achieve state-of-the-art performance for 3D semantic and instance segmentation on benchmark datasets in terms of accuracy, speed and computational cost.

II. Related Work

2D Semantic and Instance Segmentation. The great advances in semantic and instance segmentation have largely been driven by the success of fully convolutional neural networks (FCNs) [5]. Numerous approaches [10]–[16] based on FCNs have dominated semantic segmentation tasks. [4], [17], [18] learned to segment instances by proposing segmentation candidates based on the region-based CNN (R-CNN) [19]. A top-down detector-based Mask R-CNN framework was first introduced by He et al. [3] to simultaneously perform mask and class label prediction. By contrast, bottom-up methods such as [20], [21] aim to assign per-pixel predictions to instances.

3D Point Clouds Segmentation. Recent advances in deep neural networks have also led to various cutting-edge 3D semantic [22]–[28] and instance segmentation [9], [29]–[31] approaches. Using voxelized volumes to represent 3D point clouds is a popular and effective strategy. [32]–[35] transferred 3D point clouds data into regular volumetric occupancy grids and applied 3D CNNs to perform voxel-level predictions. Based on the MLP, PointNet [36] was the first to directly process raw point clouds and perform point-level predictions, demonstrating high performance on both segmentation and classification tasks. Following that pioneering work, PointNet++ [37], PointCNN [38], GB-RCU [39] and RSNet [40] were developed through investigations of the local context and hierarchical learning structures. Graph neural networks have opened up more efficient and flexible ways to handle 3D segmentation [6], [41], [42]. Recently, by advancing a joint semantic and instance learning framework, [7], [8] proposed methods that achieve superior performance on both tasks.

III. Method

As depicted in Figure 2, the framework with CFSM and $\mathcal{E}_{EMED}$ removed is the baseline method. First, point clouds of size L_P are encoded into a feature matrix $F_{SHARE} \in \mathbb{R}^{L_P \times L_F}$ by the encoder (PointNet/PointNet++). Next, two tasks separately decode the shared encoded feature for their own missions. F_{SHARE} is decoded by the semantic segmentation branch into the semantic feature matrix $F_{SEM} \in \mathbb{R}^{L_P \times L_F}$ and then outputs the semantic predictions $P_{SEM} \in \mathbb{R}^{L_P \times L_C}$, where L_C is the semantic class number. The instance segmentation branch decodes F_{SHARE} into the instance feature matrix $F_{INS} \in \mathbb{R}^{L_P \times L_F}$, which is utilized to predict the per-point instance embeddings $E_{INS} \in \mathbb{R}^{L_P \times L_E}$, where L_E denotes the length of the output embedding dimensions. These embeddings are used to calculate the Euclidean distances between points for instance clustering. During the training process, the semantic branch is supervised by cross-entropy loss and the instance branch is supervised by the instance loss following [7], and the specific loss formula is detailed in [7]. In our paper, we denote this loss as $\mathcal{E}_{INS}$. For inference, we use mean-shift clustering [43] on the instance embeddings to obtain the final instance labels. The mode of the semantic labels of the points within the same instance is assigned as the predicted semantic class.

A. CFSM

Reciprocal Feature Selection and Integration. As illustrated in Figure 2, our CFSM contains two branches: $\mathcal{C}_{I-S}$ for instance-fused semantic segmentation and $\mathcal{C}_{S-I}$ for semantic-aware instance segmentation. Both $\mathcal{C}_{I-S}$ and $\mathcal{C}_{S-I}$ can be separately integrated into baseline model, when

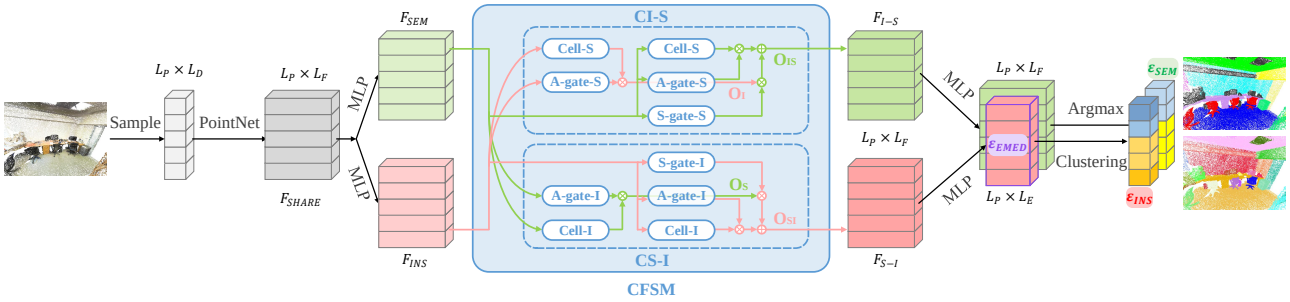

Fig. 2. Illustration of our proposed 3DCFS architecture. The randomly sampled point clouds are first input to a feed-forward network, which computes a 128-dimensional feature vector for each point. Then, the CFSM exploits and integrates the reciprocal information for semantic and instance segmentations. $\mathcal{E}_{EMED}$ is applied on the instance embeddings to help the model learn to balance the magnitudes of the embedding dimensions during training.

the other branch is replaced by the MLP. In our method, there are two types of gates: attention gates (A-gates) and selection gates (S-gates). Both are learnable modules that implement several 1×1 convolutions and activation functions. An A-gate is used for reweighting the semantic and instance features themselves before fusion, while the S-gate is used to filter or select the information from the other task. The cell is a decoding unit that contains convolutions and activation functions.

CI-S. As depicted in Figure 2, we denote F_{INS} and F_{SEM} as the instance and semantic features decoded by the MLP from F_{SHARE}. The semantic branch $\mathcal{C}_{I-S}$ contains three units: an A-gate called "A-gate-S", an S-gate named "S-gate-S" and a cell termed "Cell-S". The units with the same name share the weight parameters. For the $\mathcal{C}_{I-S}$ branch, the red F_{INS} pass through the Cell-S and A-gate-S to obtain the output O_I; the F_{SEM} in green are fed into all three units to obtain the output O_{IS}. These outputs are calculated by the dot product $\otimes$ and summation $\oplus$ operations as illustrated in Figure 2, which are formulated as follows:

$$\begin{cases} O_I = \sigma(W_{cell} * F_{INS}) \cdot \varsigma(W_A * F_{INS}) & (1) \\ O_i = \xi(W_S * F_{SEM} \cdot O_I) & (2) \\ O_{IS} = \sigma(W_{cell} * F_{SEM}) \cdot \varsigma(W_A * F_{SEM}) + O_i & (3) \\ F_{I-S} \triangleq O_{IS} & (4) \end{cases}$$

where W_{cell}, W_A and W_S are the weights of Cell-S, A-gate-S and S-gate-S, respectively, and σ, ς and ξ indicate their activation functions. The symbol $*$ denotes the convolution operation, and $\cdot$ represents the dot product. The final output of $\mathcal{C}_{I-S}$ for the semantic segmentation task is $F_{I-S} \in \mathbb{R}^{L_P \times L_F}$. Based on the attention mechanism, our A-gate has the capability to reweight the features of the task itself to facilitate extracting the crucial internal information of both tasks. Then, the representations in F_{INS}, which are useful to the semantic segmentation task, are exploited and reserved through our S-gate. This selection process is guided and controlled by the semantic features. For example, the S-gate-S in $\mathcal{C}_{I-S}$ can select features from the same instances and discover their general characteristics, which helps the model to recognize their category.

CS-I. The $\mathcal{C}_{S-I}$ architecture is the same as the $\mathcal{C}_{I-S}$ structure, except that the instance features are assisted by the semantic features. F_{SEM} is passed to $\mathcal{C}_{S-I}$ as complementary information to help improve the performance of the instance segmentation. S-gate-I in $\mathcal{C}_{S-I}$ is able to block the useless information and filter the features that blur the differences between instances as well as select more valuable representations to indicate the differences between categories for instance segmentation. The output of $\mathcal{C}_{S-I}$ for the semantic segmentation task is $F_{S-I} \in \mathbb{R}^{L_P \times L_F}$.

CFSM. As illustrated in Figure 2, $\mathcal{C}_{I-S}$ and $\mathcal{C}_{S-I}$ can be simultaneously integrated into the baseline model. As the complementary information, F_{SEM} and F_{INS} are taken as the input to the other branches $\mathcal{C}_{S-I}$ and $\mathcal{C}_{I-S}$, respectively. At the same time, F_{SEM} and F_{INS} are passed to their own branches $\mathcal{C}_{I-S}$ and $\mathcal{C}_{S-I}$, respectively.

B. Learn to Balance the Embedding Dimension Magnitude

To further improve the instance segmentation performance of our framework, we design a loss function to learn to balance the magnitudes of the output embedding dimensions. The Euclidean distance is sensitive to magnitude differences, which makes the cluster results dependent on only a few dimensions of the embedding and reduces the generalizability of the model. A traditional trick to solve this issue is to apply a mean-removal strategy on the output embeddings before instance clustering during inference. Rather than employ this post-process method, we directly apply our proposed loss function to the model to stabilize the Euclidean distance calculation during training. Specifically, we denote $\mathcal{E}_{EMED}$ as the equilibrium loss for the magnitude. The loss term can be written as follows:

$$\begin{cases} \bar{E} = \dfrac{1}{L_P} \displaystyle\sum_{i=1}^{L_P} E_i & (5) \\ \mathcal{E}_{EMED} = \dfrac{1}{L_E} \displaystyle\sum_{d=1}^{L_E} (\bar{E}_d - \mu)^2 & (6) \\ \mathcal{E}^*_{INS} = \mathcal{E}_{INS} + \alpha \mathcal{E}_{EMED} & (7) \end{cases}$$

where E_i is the embedding of each point, $\mathcal{E}^*_{INS}$ is the total instance loss of our 3DCFS, μ denotes the mean value of $\bar{E}$, and α is the balanced weight of $\mathcal{E}_{INS}$ and $\mathcal{E}_{EMED}$.

IV. Experiments

A. Datasets and Experimental Setup

Dataset. Our experiments are conducted on two benchmark datasets: the Stanford 3D Indoor Semantics Dataset (S3DIS) [44] and ShapeNet Dataset [45].

- S3DIS is a 3D scene dataset that contains large-scale scans of indoor spaces. Each point is annotated with an instance label and a semantic label from 13 semantic classes. S3DIS embeds each point into a 9-dimensional feature vector including XYZ, RGB and normalized coordinates. Following [36], we split the rooms into 1 m $\times$ 1 m overlapping blocks with stride 0.5 m on the ground plane and sample 4,096 points from each block.
- The ShapeNet part dataset contains 16,881 3D shapes from 16 semantic classes. Each point is associated with one of the 50 different parts. We utilize the instance annotations from [9] as the ground-truth labels. Each shape is represented by point clouds with 2,048 points following [36], and each point is represented by an XYZ 3-dimensional vector. The point clouds are sampled for the input of our framework following [7].

Evaluation. Following [7], we conduct experiments involving S3DIS on Area5. The performance on the sixth fold cross validation with microaveraging [4] is also measured. For semantic segmentation, we calculate the overall accuracy (oAcc), mean accuracy (mAcc) and mean IoU (mIoU). To evaluate the performance of instance segmentation, we use the coverage (Cov) and weighted coverage (WCov), the specific calculation formulas are detailed in [21], [46], [47].

Implementation Details. For instance segmentation, we train 3DCFS with $\lambda = 0.001$. We use five output embeddings following [7] and set α to 0.01. We train the network for 50 epochs for PointNet and PointNet++ with a batch size of 12 and the base learning rate set to 0.001 and divided by 2 every 300 k iterations. We select the Adam optimizer to optimize the network on a single GPU (Tesla P100) and set the momentum to 0.9 for the training process. During the inference process, we set the bandwidth to 0.6 for mean-shift clustering and apply the BlockMerging algorithm [9] to merge instances from different blocks. The code will be available at GitHub, which contains more details.

B. S3DIS

Following [7], we conducted experiments on the S3DIS dataset based on the PointNet and PointNet++ backbone networks.

Quantitative Results. For Area5, the quantitative results of 3DCFS on the instance and semantic segmentation tasks are shown in Table I. Using the PointNet backbone, 3DCFS achieves 44.4 mWCov, which dramatically outperforms the state-of-the-art method ASIS [7] by 2.8 and significantly improves the mPrec by 3.3. After replacing the backbone with PointNet++, we still achieve 4.5 mWCov gains and 2.1 mIoU gains on the instance and semantic segmentation tasks, respectively. Table II shows the performance of our 3DCFS on semantic segmentation in all 6 areas. 3DCFS outperforms ASIS by 1.4 for mWCov and 1.8 for mPrec. Using the PointNet++ backbone on all 6 areas, 3DCFS improves the mAcc by 2.3 and the mIou by 1.0. Clearly, our 3DCFS method outperforms the SOTA method ASIS [7] by a large margin. As reported in Table I and II, whether constructed upon the PointNet or PointNet++ backbones, evaluated in Area5 or 6-fold CV, our method consistently obtains better performance on both instance and semantic segmentation tasks than the state-of-the-art methods. The stable improvement demonstrates that our 3DCFS is a general and effective framework that can be built upon different network backbones. Table IV shows the instance and semantic segmentation results for specific categories. We reproduced the results of ASIS [8] and JSIS3D [7] using the code at GitHub published by the respective authors to make a full class comparison with the same PointNet backbone.

TABLE I

Instance (red) and semantic (green) segmentation results on S3DIS dataset (Test on Area5).

Backbone	Method	mCov	mWCov	mPrec	mRec
PN	SGPN [9]	32.7	35.5	36.0	28.7
	ASIS [7]	38.2	41.6	44.2	35.6
	3DCFS	**41.2**	**44.4**	**47.5**	**39.4**
PN++	ASIS [7]	44.7	47.6	54.3	43.2
	3DCFS	**49.0**	**52.1**	**55.5**	**45.9**
Backbone	Method	mAcc	mIoU	oAcc	
PN	PN [36]	52.1	43.4	83.5	/
	ASIS [7]	55.4	46.5	84.8	
	3DCFS	**56.4**	**47.1**	**84.9**	
PN++	ASIS [7]	60.9	53.4	86.9	
	3DCFS	**62.7**	**55.5**	**87.8**	

TABLE II

Instance (red) and semantic (green) segmentation results on S3DIS dataset (Test on 6-fold CV).

Backbone	Method	mCov	mWCov	mPrec	mRec
PN	SGPN [9]	37.9	40.8	38.2	31.2
	ASIS [7]	44.7	48.4	53.7	41.0
	3DCFS	**46.1**	**49.8**	**55.5**	**42.7**
PN++	ASIS [7]	51.5	54.8	62.8	47.2
	3DCFS	**53.1**	**57.1**	**63.7**	**49.1**
Backbone	Method	mAcc	mIoU	oAcc	
PN	PN [36]	60.3	48.9	80.3	/
	ASIS [7]	62.9	51.6	82.0	
	3DCFS	**63.8**	**52.3**	**82.5**	
PN++	ASIS [7]	70.1	59.3	86.2	
	3DCFS	**72.4**	**60.3**	**86.3**	

Ablation Study. The ablation study results are shown in Table III. Compared with the baseline, our 3DCFS achieves obvious improvements. We obtain 3.1 mWCov and 4.2 mPrec gains for the instance segmentation task. For semantic

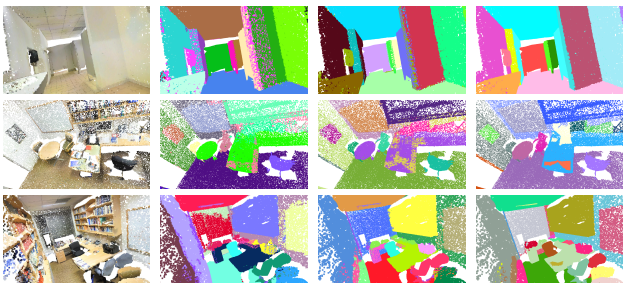

Fig. 3. Comparison of the baseline method and 3DCFS on instance segmentation. The different colors represent different instances.

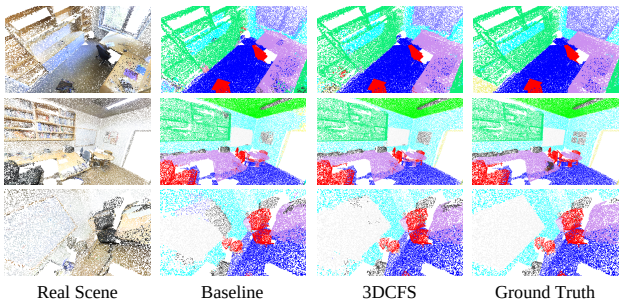

Fig. 4. Comparison of the baseline method and 3DCFS on semantic segmentation.

segmentation, we achieve 1.7 mAcc and 2.6 mIou gains. Specifically, as shown in Table III, equipped with only CI-S, our method achieves 54.4 mIoU and 62.2 mAcc, which outperforms the baseline by 1.5 and 1.2 on the semantic segmentation task, respectively. Adopting only CS-I yields 50.1 mWCov and 54.5 mPrec, which contributes to a 1.1 gain in mWCov and a 3.2 gain in mPrec compared to the baseline on the instance segmentation task. Comparing CS-I and CI-S, we find that CS-I outperforms CI-S on instance metrics, while CI-S performs better on the semantic task. The coupled module CFSM further outperforms the baseline results by a large performance margin, achieving 62.3 mAcc (semantic) and 54.7 mPrec (instance), both of which are larger than the

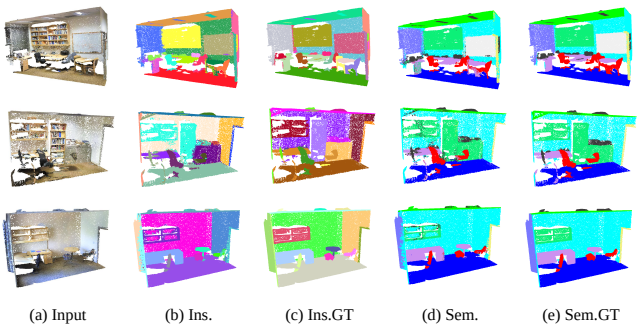

Fig. 5. Qualitative results of 3DCFS on the S3DIS test fold.

TABLE III

ABLATION STUDY ON THE S3DIS DATASET IN AREA5. CI-S REFERS TO ONLY INSTANCE FUSION; CS-I REFERS TO ONLY SEMANTIC AWARENESS; CFSM CONTAINS BOTH CI-S AND CS-I; CFSM (POST) MEANS CFSM WITH POST-PROCESS (MEAN-REMOVAL) ON INSTANCE EMBEDDING; 3DCFS IS OUR FULL APPROACH EQUIPPED WITH $\mathcal{E}_{EMED}$.

Method	mCov	mWCov	mPrec	mRec	mAcc	mIou	oAcc
Baseline	46.0	49.0	51.3	42.0	61.0	52.9	86.6
CI-S	46.7	49.6	53.9	43.1	62.2	54.4	87.4
CS-I	47.0	50.1	54.5	43.3	61.6	53.9	87.2
CFSM	48.0	50.8	54.7	44.6	62.3	54.5	87.7
CFSM (post)	48.4	51.6	**55.6**	44.0	62.4	54.6	87.5
3DCFS	**49.0**	**52.1**	55.5	**45.9**	**62.7**	**55.5**	**87.8**

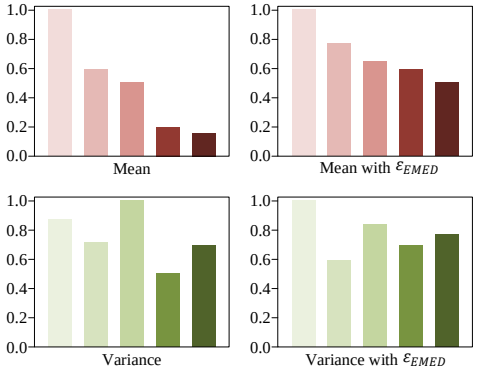

Fig. 6. Comparison of the mean and variance of 5 embedding dimensions with or without $\mathcal{E}_{EMED}$. All the values are normalized, and the mean values are ranked from high to low for better visualization.

gains provided by the individual CS-I and CI-S. The results demonstrate that improving one task can also help improve the other because each task learns better reciprocal features.

Table III also compares our $\mathcal{E}_{EMED}$ with the post-processing method mentioned in Section III-B. Figure 6 shows the comparison of the mean and variance of 5 dimensions of the embeddings with or without $\mathcal{E}_{EMED}$. The statistical analysis of the results reveals that the mean values are balanced and the variances are not influenced by $\mathcal{E}_{EMED}$, which indicates that it maintains the representational ability of the instance feature. The superior performance shows that our $\mathcal{E}_{EMED}$ successfully helps the model learn to balance the dimension magnitude. Table V shows that incorporating our proposed $\mathcal{E}_{EMED}$ boosts the performance with embedding lengths 5 and 10. Note that the improvement is much more significant as the embedding length increases.

Qualitative Results. For instance segmentation, different colors represent different instances. As depicted in Figure 3, the baseline approach incorrectly clusters two nearby different class instances into one instance (e.g., board and wall). After applying 3DCFS, the instances are correctly clustered. For semantic segmentation, each color refers to a particular class. The qualitative comparisons are shown in Figure 4. 3DCFS performs better on classifying the

TABLE IV

PER CLASS RESULTS ON THE S3DIS DATASET.

Metrics	Method	mean	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
mPrec	JSIS3D [8]	41.5	82.7	85.1	44.2	0.0	15.4	74.4	33.3	34.0	**64.3**	**20.0**	**47.5**	9.1	30.3
	ASIS [7]	53.7	**88.6**	87.5	57.8	57.1	35.7	**68.3**	59.1	**43.2**	58.4	7.3	36.0	64.8	34.4
	3DCFS	**55.3**	88.1	**89.1**	**59.3**	**63.3**	**41.1**	**68.3**	**61.2**	42.2	55.5	4.3	38.4	**73.4**	**34.6**
mAcc	SEGCloud [48]	48.9	90.1	96.1	69.9	0.0	18.4	38.4	23.1	70.4	**75.9**	**40.9**	58.4	13.0	41.6
	JSIS3D [8]	50.5	**96.7**	**99.1**	**90.0**	0.0	0.1	53.0	10.2	64.3	71.2	35.3	**64.6**	17.3	54.5
	ASIS [7]	62.9	95.6	91.5	88.4	62.4	37.1	**57.8**	67.0	**77.5**	55.7	23.2	53.1	43.0	65.2
	3DCFS	**64.7**	94.9	92.1	88.4	**64.7**	**44.8**	56.7	**69.2**	74.2	56.4	34.3	53.1	**45.8**	**66.8**

TABLE V

ABLATION STUDY ON THE S3DIS DATASET IN AREA5. CFSM5 AND CFSM10 REPRESENT THE INSTANCE SEGMENTATION WITH OUTPUT EMBEDDING LENGTH OF 5 AND 10, RESPECTIVELY. 3DCFS5 AND 3DCFS10 DENOTE THE METHOD EQUIPPED WITH $\mathcal{E}_{EMED}$.

Method	mCov	mWCov	mPrec	mRec	mAcc	mIou	oAcc
CFSM5	48.0	50.8	54.7	44.6	62.3	54.5	87.7
3DCFS5	49.0	52.1	55.5	45.9	62.7	55.5	87.8
CFSM10	45.4	48.5	52.2	40.5	61.2	53.6	87.3
3DCFS10	48.4	51.7	55.0	44.1	63.1	55.2	87.5

entire semantic information, especially at the boundaries of different categories. Figure 5 shows qualitative examples of 3DCFS on both instance and semantic segmentation. Our results are essentially the same as the ground truth, especially for instance segmentation.

Speed and Computing Resources. Table VI shows a comparison of the memory cost and computation time measured on a single GTX 1080 GPU. For a fair comparison, we conducted the experiments in the same environment, including the same GPU, batch size (4) and data (Area5). Note that all time units are minutes and all memory units are MB. In the training process, the result is the time and memory cost for one epoch. Our approach takes only 26.4 minutes and 2,227 MB, which is significantly faster and more efficient than the state-of-the-art methods. In the test process, the results show the resource consumption for inferencing. Here, our method is also found to be superior to the state-of-the-arts in terms of accuracy, speed and computational cost.

TABLE VI

COMPARISONS OF COMPUTATION SPEED, GPU MEMORY AND PERFORMANCE. THE UNITS FOR TIME AND MEMORY ARE MINUTES AND MB RESPECTIVELY.

Metrics / Method	Train		Test		mWconv
	time	memory	time	memory	
SGPN [9]	59.3	7549	209.5	420	35.5
ASIS [7]	64.7	4275	54.2	1235	40.3
3DCFS	**26.4**	**2227**	**36.3**	**307**	**44.4**

C. ShapeNet

We conducted experiments on the ShapeNet dataset using instance segmentation annotations generated by [9]. For instance segmentation, only the qualitative results are provided following [9] because no true ground truth exists. As shown in Figure 7, the tires of the car and legs of the chair and the table are properly grouped into individual instances. Both the semantic and instance segmentation results are accurate and clear. The semantic segmentation results are shown in Table VII. Our 3DCFS further outperforms the state-of-the-art method SGPN by 1.8 mIoU based on PointNet++. These results reveal that our proposed 3DCFS also has the capability to boost part segmentation performance.

TABLE VII

SEMANTIC SEGMENTATION RESULTS ON SHAPENET DATASETS.

Method	mIoU
PointNet [37]	84.3
ASIS [7]	85.0
SGPN [9]	85.8
3DCFS	**87.6**

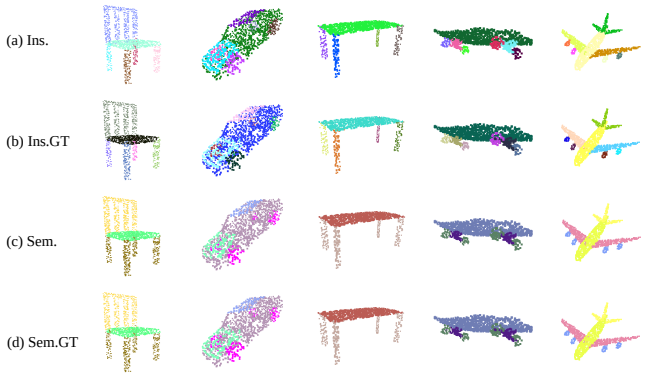

Fig. 7. Qualitative results of 3DCFS on the ShapeNet test split. (a) Instance segmentation results of 3DCFS. (b) Generated ground truth for instance segmentation. (c) Semantic segmentation results of 3DCFS. (d) Semantic segmentation ground truth.

V. CONCLUSIONS

In this paper, we proposed a fast and robust joint 3D semantic-instance segmentation framework. A novel CFSM was introduced to exploit the reciprocal information from two different tasks in a coupled manner. We also proposed a novel loss function that helped our 3DCFS learn to balance the magnitudes of the instance embedding dimensions to make the Euclidean distance calculation more reliable. Experimental results on the S3DIS and ShapeNet part datasets demonstrated the effectiveness and efficiency of 3DCFS.

REFERENCES

[1] S. James, A. J. Davison, and E. Johns, "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task," *arXiv preprint arXiv:1707.02267*, 2017.

[2] L. Du, J. Li, X. Ye, and X. Zhang, "Weakly supervised deep depth prediction leveraging ground control points for guidance," *IEEE Access*, vol. 7, pp. 5736–5748, 2018.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[4] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," *arXiv preprint arXiv:1708.02551*, 2017.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[6] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4558–4567.

[7] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," *arXiv preprint arXiv:1902.09852*, 2019.

[8] Q.-H. Pham, D. T. Nguyen, B.-S. Hua, G. Roig, and S.-K. Yeung, "Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields," *arXiv preprint arXiv:1904.00699*, 2019.

[9] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2569–2578.

[10] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[11] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[12] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Exploring context with deep structured models for semantic segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1352–1366, 2018.

[13] H. Li, P. Xiong, H. Fan, and J. Sun, "Dfanet: Deep feature aggregation for real-time semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9522–9531.

[14] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2604–2613.

[15] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7519–7528.

[16] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 982–991.

[17] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 534–549.

[18] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems*, 2015, pp. 1990–1998.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[20] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *Advances in Neural Information Processing Systems*, 2017, pp. 2277–2287.

[21] S. Liu, J. Jia, S. Fidler, and R. Urtasun, "Sgn: Sequential grouping networks for instance segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3496–3504.

[22] J. Guerry, A. Boulch, B. Le Saux, J. Moras, A. Plyer, and D. Filliat, "Snapnet-r: Consistent 3d multi-view semantic labeling for robotics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 669–678.

[23] B. Shi, S. Bai, Z. Zhou, and X. Bai, "Deeppano: Deep panoramic representation for 3-d shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[24] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[25] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5648–5656.

[26] A. Komarichev, Z. Zhong, and J. Hua, "A-cnn: Annularly convolutional neural networks on point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7421–7430.

[27] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," *arXiv preprint arXiv:1811.07246*, 2018.

[28] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, "3d recurrent neural networks with context fusion for point cloud semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 403–417.

[29] J. Lahoud, B. Ghanem, M. Pollefeys, and M. R. Oswald, "3d instance segmentation via multi-task metric learning," *arXiv preprint arXiv:1906.08650*, 2019.

[30] J. Hou, A. Dai, and M. Nießner, "3d-sis: 3d semantic instance segmentation of rgb-d scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4421–4430.

[31] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "Gspn: Generative shape proposal network for 3d instance segmentation in point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3947–3956.

[32] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[33] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[34] J. Huang and S. You, "Point cloud labeling using 3d convolutional neural network," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2670–2675.

[35] G. Riegler, A. Osman Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3577–3586.

[36] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[37] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[38] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in Neural Information Processing Systems*, 2018, pp. 820–830.

[39] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3d semantic segmentation of point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 716–724.

[40] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3d segmentation of point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2626–2635.

[41] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *arXiv preprint arXiv:1801.07829*, 2018.

[42] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 296–10 305.

[43] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 603–619, 2002.

[44] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.

[45] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, *et al.*, "A scalable active framework for region annotation in 3d shape collections," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 210, 2016.

[46] M. Ren and R. S. Zemel, "End-to-end instance segmentation with recurrent attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6656–6664.

[47] W. Zhuo, M. Salzmann, X. He, and M. Liu, "Indoor scene parsing with instance segmentation, semantic labeling and support relationship inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5429–5437.

[48] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 537–547.